CLAIMS

Listing of Claims:

1. (Currently Amended) A method including:

dedicating a first portion comprising one or more ways of an ~~resource~~ N way set associative cache exclusively to a first thread;

dedicating a second portion comprising one or more ways of the ~~resource~~ cache exclusively to a second thread; ~~and~~

dynamically sharing a third portion comprising one or more ways of the ~~resource~~ cache between the first and second threads; and

performing victim selection in the cache by,

examining a Least Recently Used (LRU) history for a selected set to identify a least recently used way within the cache as a candidate way to store an information item associated with the first thread;

determining whether the candidate way is within the first or the third portion of the cache;

if the candidate way is within the first or the third portion of the cache, then storing the information associated with the first thread in the candidate way; and

if the candidate way is within the second portion of the cache, then identifying a further way within the cache as being the candidate way, wherein the further way is not the candidate way previously identified.

2. (Currently amended) The method of claim 1 wherein the dynamic sharing of the third portion of the ~~resource~~cache is performed according to resource demands of the respective first and second threads.

3. (Canceled)

4. (Canceled)

5. (Canceled)

6. (Currently Amended) The method of ~~claim 4~~ claim 1wherein the examining includes examining the LRU history to identify ~~the identification of the selected way within the selected set as the candidate way comprises identifying a~~ M least recently used ways within the ~~select set~~cache, wherein M>1 ~~that was least recently used~~, and wherein the determining is performed for the M candidate ways, and wherein the further way is another of the M candidate ways.

7. (Currently Amended) The method of claim 1~~6~~ wherein the identifying of the further way as being the candidate way ~~identification of the selected way within the selected set as the candidate way~~ comprises identifying a ~~way within the selected set that was~~ second-least recently used way within the cache.

8. (Canceled)

9. (Currently Amended) The method of claim 1~~8~~ ~~including~~ wherein the examining includes examining ~~a set of~~ entries within the LRU history for the selected set, each entry ~~within the set of entries indicating~~ corresponding to a respective one of the ways ~~within the selected set~~, wherein the ~~set of~~ entries ~~is~~ are ordered in a sequence determined by least recent usage of the~~a~~ respective ways ~~and the selection of the candidate way comprises performing a sequential examination of the entries of the set of entries to locate a least recently used way that comprises either the first or the second way~~.

10. (Previously Presented) The method of claim 14 wherein ~~memory~~ the cache is ~~comprises~~ a trace cache memory, and wherein the information item associated with the first thread comprises a microinstruction of the first thread.

11. (Currently Amended) An ~~resource~~ N way set associative cache comprising:

a first portion comprising one or more ways dedicated to utilization by a first thread executing within a multi-threaded processor;

a second portion comprising one or more ways dedicated to utilization by a second thread executing within the multi-threaded processor; ~~and~~

a third portion comprising one or more ways shared by the first and second threads; and

selection logic to examine a Least Recently Used (LRU) history for a selected set to identify a least recently used way within the cache as a candidate way within which to store an information item associated with the first thread, to store the information associated the first thread in the candidate way if the candidate way is within the first or third portions of the cache, but if the candidate way is within the second portion of the cache, then to identify a further way within the cache as being the candidate way, wherein the further way is not the candidate way previously identified.

12. (Canceled)

13. (Canceled)

14. (Canceled)

15. (Currently Amended) The ~~resource~~ cache of claim ~~12~~ 11 wherein ~~the selection logic identifies a further way within the selected set of the memory as the candidate way to receive the information item associated with the first thread if the selected way comprises the second way~~the third portion of the cache is shared according to resource demands of the first and second threads.

16. (Currenlty Amended) The ~~resource~~ cache of claim ~~14~~ 11 wherein the selection logic ~~identifies the selected way within the selected set as the candidate way by identifying the selected way within the select set as a last recently used way within the selected set~~is to examine for the selected set the LRU history to identify M least recently used ways within the cache, and wherein M>1.

17. (Currently amended) The ~~resource~~ cache of claim 11~~5~~ wherein the selection logic identifies as the further way ~~within the selected set a candidate way by identifying the further way within the selected set as~~ a second-least recently used way within the ~~selected set~~cache.

18. (Canceled)

19. (Currently Amended) The ~~resource~~ cache of claim 11~~8~~ wherein the selection logic examines ~~a set of~~ entries within the LRU history for the selected set to identify the least recently used way~~, each entry within the set of entries indicating a respective way within the selected set~~, wherein the ~~set of~~ entries ~~is~~ are ordered in a sequence determined by least recent usage of the ways to which they correspond~~a respective way and the selection of the candidate way comprises performing a sequential examination of the entries of the set of entries to locate a least recently used way that comprises either the first or the second way~~.

20. (Currently Amended) The ~~resource~~ cache of claim 11~~8~~ wherein the ~~memory~~ cache ~~comprising~~ is a trace cache memory, and wherein the information item associated with the first thread comprises a microinstruction of the first thread.

21. (Canceled)

22-23. (Canceled)

24. (Currently Amended) A method including:

~~defining~~ configuring an ~~memory resource~~N way set associative cache, associated with a multi-threaded processor, to include first and second portions comprising one or more ways dedicated to the first and second threads respectively and a third portion comprising one or more ways shared between the first and second threads;

for an information item associated with the first thread, examining for a selected set a history of least recently used ~~portions~~ ways ~~to identify either the first portion or the third portion as being a least recently used portion available to the first thread~~until one is found that is within one of the first and third portions; and

storing the information item within the ~~least recently used portion~~found way.

25. (Currently Amended) The method of claim 24 wherein~~,~~ the sharing of the third portion of the cache is performed according to resource demands of the respective first and second threads~~for the information item associated with the first thread, the second portion is excluded from the identification as the least recently used portion on account of being dedicated to the second thread~~.

26. (Currently Amended) The method of claim 24, wherein the examining is performed on M ways of the history of least recently used ways, and where M>1~~memory resource comprises a N way set associative cache memory and wherein the first, second and third portions comprising respective first, second and third ways~~.

27. (Currently Amended) The method of claim 26 wherein the examination of the history of least recently used ~~portions~~ ways includes examining entries within the~~a~~ ~~least recently used~~

history, each entry corresponding to a respective one of the ways, wherein the entries are ordered in a sequence determined by least recent usage of the respective ways ~~for a selected set of the set associative cache memory~~.

28. (Currently Amended) The method of claim 24 wherein the cache ~~memory comprises~~ is a trace cache memory, and wherein the information item associated with the first thread comprises a microinstruction of the first thread.

29. (Currently Amended) A computer ~~readable medium~~storage storing a sequence of instructions that, when executed within a processor, causes the processor to perform the steps of:

dedicating a first portion comprising one or more ways of a N way set associative cache ~~resource~~ exclusively to a first thread;

dedicating a second portion comprising one or more ways of the ~~resource~~ cache exclusively to a second thread; ~~and~~

dynamically sharing a third portion comprising one or more ways of the ~~resource~~ cache between the first and second threads; and

performing victim selection in the cache by,

examining a Least Recently Used (LRU) history for a selected set to identify a least recently used way within the cache as a candidate way to store an information item associated with the first thread;

determining whether the candidate way is within the first or the third portion of the cache;

if the candidate way is within the first or the third portion of the cache, then

storing the information associated with the first thread in the candidate way; and

if the candidate way is within the second portion of the cache, then identifying a

further way within the cache as being the candidate way, wherein the further way is not the candidate way previously identified.

30. (Currently Amended) The computer ~~readable~~ storage ~~medium~~ of claim 29 wherein the dynamic sharing of the third portion of the ~~resources~~ cache is performed according to resource demands of the respective first and second threads.

31. (Currently Amended) The computer storage of claim 29 wherein the examining includes examining the LRU history to identify M least recently used ways within the cache, wherein M>1 , and wherein the determining is performed for the M candidate ways, and wherein the further way is another of the M candidate ways.

~~A processor comprising:~~

~~a cache divided into a plurality of partitions;~~

~~victim selection logic to,~~

~~allow replacement of a first information element from a first thread in a shared partition of the cache with a second information element from a second thread but to prevent replacement of a third information element from the first thread in a first partition of the cache with the second information element from the second thread; and~~

~~allow replacement of a fourth information element from the second thread in the shared partition of the cache with a fifth information element from the first thread but to prevent replacement of a sixth information element from the second thread in a second partition of the cache with the fifth information element from the first thread.~~

32. (Currently Amended) The computer storage of claim 29 wherein the identifying of the further way as being the candidate way comprises identifying a second-least recently used way within the cache~~The processor of claim 31 wherein the first partition is a first thread partition, the second partition is a second thread partition.~~

33. (Currently Amended) The computer storage of claim 29 wherein the examining includes examining entries within the LRU history for the selected set, each entry corresponding to a respective one of the ways, wherein the entries are ordered in a sequence determined by least recent usage of the respective ways ~~The processor of claim 31, wherein the cache is a trace cache and the processor is capable of fine simultaneous multithreading~~.

34. (Currently Amended) The computer storage of claim 29 wherein the cache is a trace cache memory, and wherein the information item associated with the first thread comprises a microinstruction of the first thread ~~The apparatus of claim 31, wherein the cache is a multi-way set associative cache and each of the partitions includes one or more ways of the cache~~.

35. (Canceled)

36. (Currently Amended) A method comprising:

detecting misses in an N way set associative cache of a processor;

performing victim selection responsive to each of the misses in a manner that partitions the capacity of the cache to make a first and second portion each comprising ~~partition~~ one or more ways available for replacement respectively only by each of a first and second instruction threads while at the same time that defines a shared portion comprising one or more ways ~~of the capacity of the cache that is~~ available to both the first and second instruction threads, wherein the performing victim selection responsive to each of the misses includes examining entries of a least recently used history until an available one of said ways within a selected set is found.

37. (Currently Amended) The method of claim 36, further comprising:

performing fine multithreading of the first and second threads in the processor.

38. (Currently Amended) The method of claim 36, wherein the examining ~~performing victim selection~~ includes ~~determining~~ examining M of the entries corresponding to the M ~~which of a plurality of entries in the cache is the~~ least recently used of the ways, and wherein M>1.

39. (Currently Amended) An apparatus comprising:

a processor including,

an N way set associative cache having a storage capacity, and

victim selection logic including logic to partition the storage capacity of the cache ~~with a bias toward~~ having a dedicated portion comprising one or more ways for each of a first and second instruction threads while at the same time having a shared portion comprising one or more ways accessible by both the first and second threads, wherein the victim selection logic responsive to each of the misses in the cache to examine entries of a least recently used history for a selected set of the cache until one of said ways is found that is available to the instruction thread causing that cache miss~~;~~ ~~and~~

~~a main memory coupled to the processor having stored therein the first and second instructions threads, which are of a multi-media type~~.

40. (Previously Presented) The apparatus of claim 39, wherein the cache is a trace cache and the processor is capable of fine simultaneous multithreading.

41. (Currently Amended) The apparatus of claim 39, wherein the third portion is dynamically shared according to resource demands of the respective first and second threads~~the cache is a multi-way set associative cache and each of the partitions includes one or more ways of the cache~~.

42. (Currently Amended) The apparatus of claim 39, wherein ~~the logic to partition is based on a least recently used scheme~~the cache is a trace cache, wherein the trace cache is to stored microinstructions of the first and second threads.

43. (New) The apparatus of claim 39 wherein the entries are ordered in a sequence determined by least recent usage of the ways to which they correspond.

44. (New) The apparatus of claim 39 wherein the victim selection logic responsive to each of the misses in the cache to examine M entries corresponding to M least recently used ways within the cache, and wherein M>1.

45. (New) The apparatus of claim 39 wherein the shared portion of the cache is shared according to resource demands of the respective first and second threads.

46. (New) The method of claim 36 wherein the shared portion of the cache is shared according to resource demands of the respective first and second threads.

47. (New) The method of claim 36 wherein each of the entries corresponds to a respective one of the ways, and wherein the entries are ordered in a sequence determined by least recent usage of the respective ways .

48. (New) The method of claim 36 wherein the cache is a trace cache memory to store microinstructions of the first and second instruction threads.